Deep Learning - Week 7

1. Which of the following statements about L2 regularization is true?

   (a) It adds a penalty term to the loss function that is proportional to the absolute value of the weights.

   (b) It results in sparse solutions for w.

   (c) It adds a penalty term to the loss function that is proportional to the square of the weights.

   (d) It is equivalent to adding Gaussian noise to the weights.

   Correct Answer: (c)
   Solution:

   It adds a penalty term to the loss function that is proportional to the square of the weights. L2 regularization, also known as **Ridge Regularization**, adds a penalty term to the loss function that is proportional to the sum of the squares of the weights. The modified loss function typically looks like:

$$L_{\text{reg}} = L + \lambda \sum w^2$$

   where λ is a hyperparameter that controls the strength of regularization.

   Now, let's analyze the other options:

   It adds a penalty term to the loss function that is proportional to the absolute value of the weights. Incorrect. This describes **L1 regularization (Lasso)**, not L2.

   It results in sparse solutions for w. Incorrect. L2 regularization does **not** lead to sparse solutions (i.e., it does not force weights to be exactly zero). Instead, it **shrinks** weights toward zero but usually keeps them nonzero. **L1 regularization** is the one that encourages sparsity.

   It is equivalent to adding Gaussian noise to the weights. Incorrect. While L2 regularization can be interpreted as **a prior in a Bayesian framework** (i.e., assuming a Gaussian prior on weights), it does **not** mean that Gaussian noise is explicitly added to the weights during training.

   Common Data Q2-Q3

   Consider two models:

$$\hat{f}_1(x) = w_0 + w_1 x$$

$$\hat{f}_2(x) = w_0 + w_1 x^2 + w_2 x^2 + w_4 x^4 + w_5 x^5$$

2. Which of these models has higher complexity?

   (a) $\hat{f}_1(x)$

   (b) $\hat{f}_2(x)$

(c) It is not possible to decide without knowing the true distribution of data points in the dataset.

Correct Answer: (b)
Solution: Model $\hat{f}_2(x)$ has higher complexity compared to Model $\hat{f}_1(x)$. The complexity of a model generally increases with the degree of the polynomial terms. Model $\hat{f}_1(x)$ is a linear model, whereas Model $\hat{f}_2(x)$ includes higher-degree polynomial terms (specifically x^2 and x^5), making it capable of capturing more complex patterns. Therefore, $\hat{f}_2(x)$ is more complex.

3. We generate the data using the following model:

$$y = 7x^3 + 12x + x + 2.$$

We fit the two models $\hat{f}_1(x)$ and $\hat{f}_2(x)$ on this data and train them using a neural network.

(a) $\hat{f}_1(x)$ has a higher bias than $\hat{f}_2(x)$.
(b) $\hat{f}_2(x)$ has a higher bias than $\hat{f}_1(x)$.
(c) $\hat{f}_2(x)$ has a higher variance than $\hat{f}_1(x)$.
(d) $\hat{f}_1(x)$ has a higher variance than $\hat{f}_2(x)$.

Correct Answer: (a),(c)
Solution: $\hat{f}_1(x)$ has a higher bias than $\hat{f}_2(x)$. (Because $\hat{f}_1(x)$ is simpler and cannot capture the true complexity of the data.) $\hat{f}_2(x)$ has a higher variance than $\hat{f}_1(x)$. (Because $\hat{f}_2(x)$ is more complex and may fit the training data too closely.)

4. Suppose that we apply Dropout regularization to a feed forward neural network. Suppose further that mini-batch gradient descent algorithm is used for updating the parameters of the network. Choose the correct statement(s) from the following statements.

(a) The dropout probability p can be different for each hidden layer
(b) Batch gradient descent cannot be used to update the parameters of the network
(c) Dropout with $p = 0.5$ acts as a ensemble regularize
(d) The weights of the neurons which were dropped during the forward propagation at t^{th} iteration will not get updated during $t + 1^{th}$ iteration

Correct Answer: (a),(c)
Solution:

(a) **The dropout probability p can be different for each hidden layer:**
- *True.* It is common practice to apply different dropout rates to different hidden layers, which allows for more control over the regularization strength applied to each layer.

(b) **Batch gradient descent cannot be used to update the parameters of the network:**

- *False.* Batch gradient descent, as well as mini-batch gradient descent, can be used to update the parameters of a network with dropout regularization. Dropout affects the training phase by randomly dropping neurons but does not prevent the use of gradient descent algorithms for parameter updates.

(c) **Dropout with $p = 0.5$ acts as an ensemble regularizer:**

- *True.* Dropout with $p = 0.5$ can be seen as an ensemble method in the sense that, during training, different subsets of neurons are active, which can be interpreted as training a large number of "thinned" networks. During testing, the full network is used but with the weights scaled to account for the dropout, effectively acting as an ensemble of these thinned networks.

(d) **The weights of the neurons which were dropped during the forward propagation at t-th iteration will not get updated during $t + 1$-th iteration:**

- *False.* During training, dropout randomly drops neurons in each mini-batch iteration, but this does not mean that the weights of dropped neurons are not updated. The update process occurs based on the backpropagation of the loss through the network, and weights are updated according to the gradients computed from the dropped and non-dropped neurons.

5. We have trained four different models on the same dataset using various hyperparameters. The training and validation errors for each model are provided below. Based on this information, which model is likely to perform best on the test dataset?

Model	Training error	Validation error
1	0.8	1.4
2	2.5	0.5
3	1.7	1.7
4	0.2	0.6

(a) Model 1

(b) Model 2

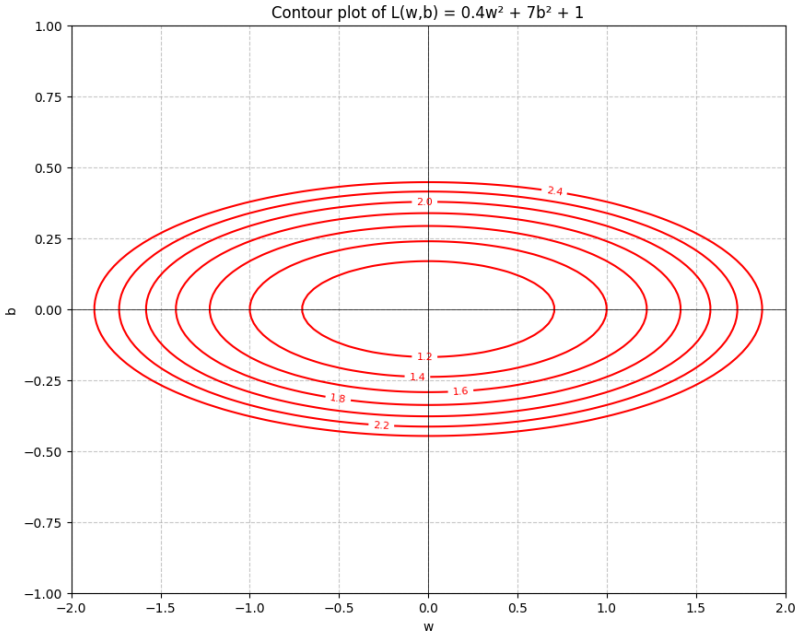
(c) Model 3

(d) Model 4

Correct Answer: (d)
Solution: Model 4 has both low training loss and low validation loss. Hence Model 4 will give you best results.

Common Data Q6-Q9

Consider a function $L(w, b) = 0.4w^2 + 7b^2 + 1$ and its contour plot given below:



6. What is the value of $L(w^*, b^*)$ where w^* and b^* are the values that minimize the function.

Correct Answer: 1

Solution: To find the value of $L(w^*, b^*)$ where w^* and b^* are the values that minimize the function

$$L(w, b) = 0.4w^2 + 7b^2 + 1,$$

We follow these steps:

1. Find the Minimum Values of w and b:

The partial derivatives of L with respect to w and b are:

$$\frac{\partial L}{\partial w} = 0.8w$$
$$\frac{\partial L}{\partial b} = 14b$$

Setting these partial derivatives to zero:

$$0.8w = 0 \implies w = 0$$
$$14b = 0 \implies b = 0$$

Therefore, the values that minimize the function are $w^* = 0$ and $b^* = 0$.

2. Evaluate L at w^* and b^*:

Substitute $w^* = 0$ and $b^* = 0$ into the function $L(w, b)$:

$$L(w^*, b^*) = L(0, 0) = 0.4(0)^2 + 7(0)^2 + 1 = 1$$

Thus, the value of $L(w^*, b^*)$ is 1.

7. What is the sum of the elements of $\nabla L(w^*, b^*)$?

   Correct Answer: 0
   Solution: The gradient $\nabla L(w, b)$ is:

$$\nabla L(w, b) = \left(\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b} \right) = (0.8w, 14b)\,.$$

   At $w^* = 0$ and $b^* = 0$, the gradient is:

$$\nabla L(w^*, b^*) = (0, 0)\,.$$

   The sum of the elements of $\nabla L(w^*, b^*)$ is:

$$0 + 0 = 0.$$

8. What is the determinant of $H_L(w^*, b^*)$, where H is the Hessian of the function?

   Correct Answer: 11.2
   Solution: The Hessian matrix $H_L(w, b)$ is:

$$H_L(w, b) = \begin{bmatrix} \frac{\partial^2 L}{\partial w^2} & \frac{\partial^2 L}{\partial w \partial b} \\ \frac{\partial^2 L}{\partial b \partial w} & \frac{\partial^2 L}{\partial b^2} \end{bmatrix}.$$

   Compute the second-order partial derivatives:

$$\frac{\partial^2 L}{\partial w^2} = 0.8$$
$$\frac{\partial^2 L}{\partial b^2} = 14$$
$$\frac{\partial^2 L}{\partial w \partial b} = \frac{\partial^2 L}{\partial b \partial w} = 0$$

   Thus, the Hessian matrix is:

$$H_L(w, b) = \begin{bmatrix} 0.8 & 0 \\ 0 & 14 \end{bmatrix}.$$

   The determinant of this matrix is:

$$\text{Determinant} = (0.8 \cdot 14) - (0 \cdot 0) = 11.2.$$

9. Compute the Eigenvalues and Eigenvectors of the Hessian. According to the eigenvalues of the Hessian, which parameter is the loss more sensitive to?

(a) b

(b) w

Correct Answer: (a)
Solution: The Hessian matrix is:

$$H_L(w, b) = \begin{bmatrix} 0.8 & 0 \\ 0 & 14 \end{bmatrix}.$$

The eigenvalues are $\lambda_1 = 0.8$ and $\lambda_2 = 14$, with corresponding eigenvectors $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, respectively. The larger eigenvalue $\lambda_2 = 14$ corresponds to the parameter b.

10. Consider the problem of recognizing an alphabet (in upper case or lower case) of English language in an image. There are 26 alphabets in the language. Therefore, a team decided to use CNN network to solve this problem. Suppose that data augmentation technique is being used for regularization. Then which of the following transformation(s) on all the training images is (are) appropriate to the problem

(a) Rotating the images by $\pm 10^\circ$

(b) Rotating the images by $\pm 180^\circ$

(c) Translating image by 1 pixel in all direction

(d) Cropping

Correct Answer: (a),(c),(d)
Solution:

Cropping:

Appropriate. Cropping is useful for augmenting data by varying the parts of the image that are used for training. This can help the model learn to recognize letters even if they are partially obscured or not centered perfectly. It ensures that the model is robust to variations in the position of the letter within the image.

Rotating the images by $\pm 10^\circ$:

Appropriate. Rotating images slightly (such as $\pm 10^\circ$) helps the model become invariant to small rotational changes. This is useful because in practical scenarios, characters might be slightly tilted, and the model should be able to recognize them regardless of minor rotations.

Rotating the images by 180°:

Not Appropriate. Rotating images by 180° is generally not useful for character recognition because it might lead to images that are completely inverted. For example, 'A' would become 'Λ' and 'B' would become 'q'. Such rotations do not usually represent valid variations in the context of character recognition.

Translating the image by 1 pixel in all directions:

Appropriate. Translating images by small amounts (such as 1 pixel) helps the model become robust to slight positional shifts. This can improve the model's ability to recognize characters that are not perfectly aligned or are slightly shifted.